



# Développement d'applications mobiles

M2 Miage Promo 2020/2021



Ouzamou Adil  
Remaud Rémi

## Introduction:

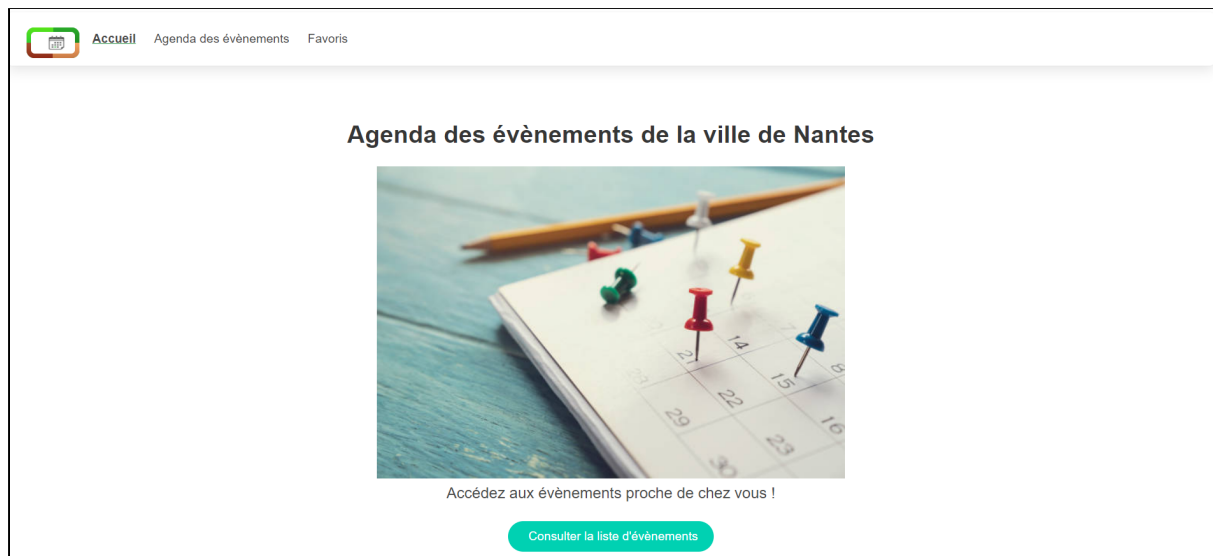
Nous avons réalisé une application VueJS dans le cadre du TP de développement d'application Mobile. Cette application consiste en un agenda des événements de la ville de Nantes. L'application récupère les événements via une API basée sur les informations contenues sur ce [site](#). Un .APK et un code android source de l'application est disponible [ici](#), avec le code source, pour avoir une version mobile de l'application.

## Présentation de l'application:

L'application que nous avons implémenté permet de lister l'ensemble des événements de la ville de Nantes. Pour ce faire, l'application est décomposée en 4 pages différentes :

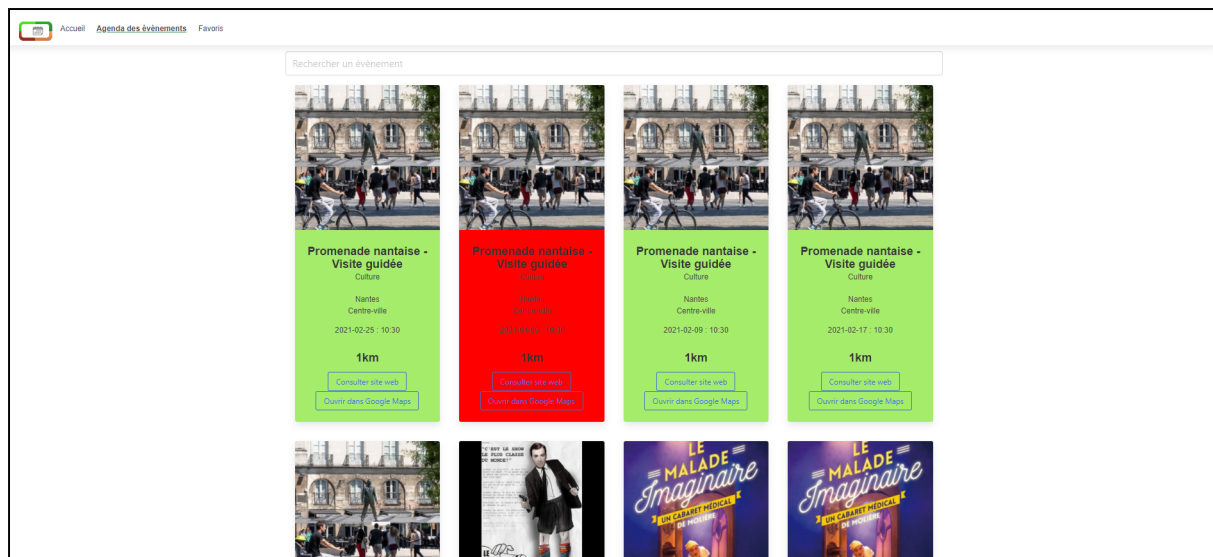
1. Page d'accueil
2. Agenda des événements (Listing des événements)
3. Détail d'un événement
4. Page des favoris

On peut voir ci-dessous la page d'accueil de l'application, celle-ci apparaît au lancement de l'application ou via le bouton **Accueil** présent sur le menu.



### 1ère page : HomePage.Vue

Ensuite, on retrouve la page **Agenda des événements** qui est accessible depuis la page d'accueil ou via le menu également.

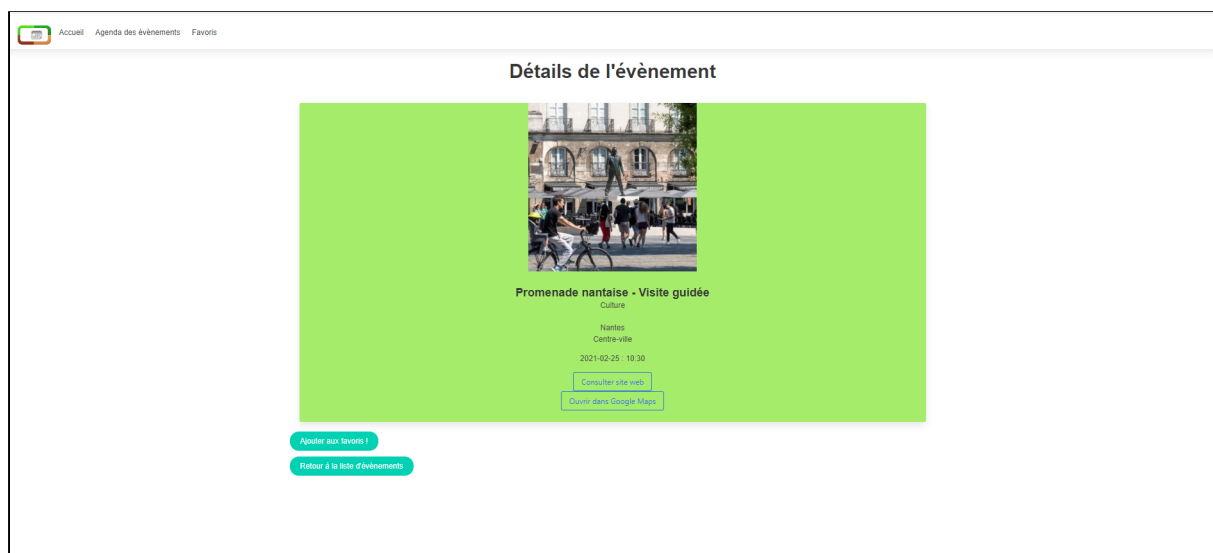


### 2ème page : ListEvents.View

Via cette page, plusieurs actions sont possibles pour un utilisateur. Il peut :

1. Consulter la page web de l'événement
2. Ouvrir l'adresse de l'événement sur Google Maps
3. Ouvrir le détail d'un événement en cliquant sur l'événement voulu
4. Effectuer une recherche dans la barre de recherche pour filtrer les événements affichés

Le **détail d'un événement** se présente comme ceci :

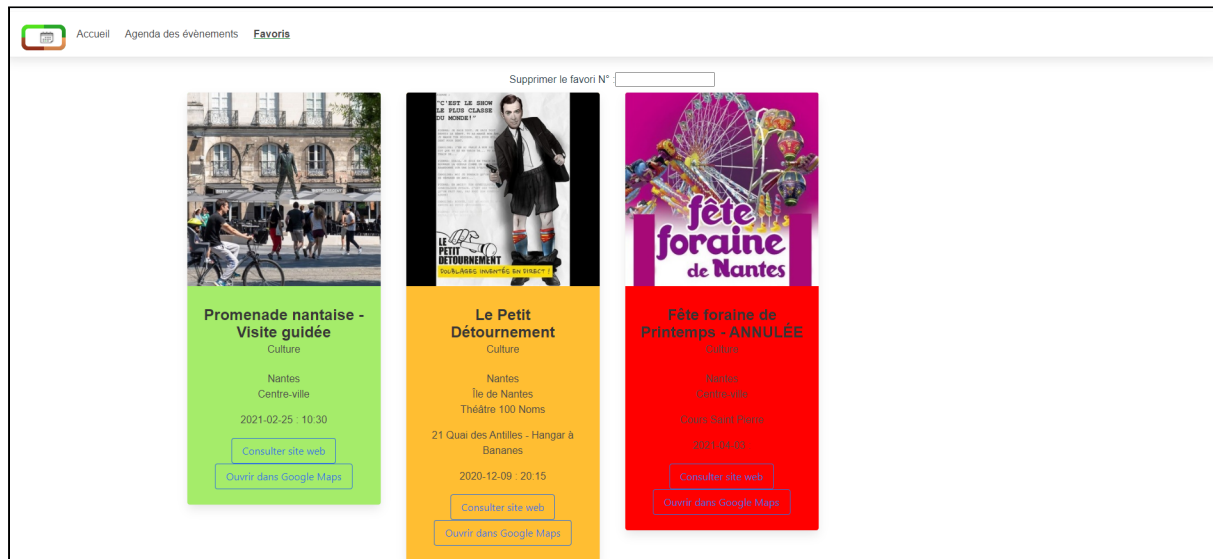


### 3ème page : EventDetailPage.View

Le détail va permettre d'avoir uniquement une vue centrée sur un seul événement et d'avoir la possibilité d'ajouter celui-ci aux favoris. De plus, l'ajout aux favoris fera apparaître une notification pour s'assurer que l'événement a bien été ajouté. Cet ajout vous redirigera sur la page **favoris**.

Cette page de détail contrairement aux autres, n'est pas accessible via le menu mais uniquement depuis la page **Agenda des événements**.

Pour finir, nous avons la page **Favoris** qui permet à l'utilisateur de pouvoir retrouver facilement tous les événements qu'il aura ajouté à ses favoris en amont.



#### 4ème page : Favoris.Vue

On retrouve également sur cette page la possibilité pour l'utilisateur de **supprimer un événement de ses favoris**. Pour ce faire, il suffit de renseigner le numéro de l'événement à supprimer (**numéro compris entre 1 et le nombre d'événements présents dans les favoris**) et d'appuyer sur entrée pour valider le choix. Cette suppression vous redirigera vers la page **Agenda des événements** pour vous permettre d'en choisir un nouveau si vous le souhaitez

#### Choix d'implémentations:

- **Gestion des favoris:** Variable globale
  - Nous avons fait le choix de créer un fichier **variable.js** contenant une variable **ArrayOfId** de type **Array()**. Celle-ci sert à stocker l'ensemble des id qui nous seront nécessaire pour savoir quels événements sont à afficher dans la page favoris. Cela évite que la variable soit réinitialisée lors des divers changements de page et nous garantit de la fiabilité des valeurs contenues dans cette variable.
- Nous avons choisi d'implémenter la **gestion des routes** suivante:
  - Au lancement de l'application l'utilisateur arrive sur la page d'accueil, il peut ensuite se diriger vers la liste des événements.
  - Une fois sur cette liste, il peut cliquer sur un événement pour afficher son détail.
  - Sur la page de détail d'un événement, l'utilisateur peut ajouter un favoris et sera redirigé vers sa page de favoris. L'utilisateur peut aussi revenir sur la page contenant la listes des événements via les 2 pages précédentes (Détail et Favoris)
  - A tout moment, l'utilisateur peut se déplacer vers les pages Accueil, Agenda des événements et Favoris via la barre de menu en haut de la page.
- **Un événement est composé des informations** suivantes:

- une image décrivant l'événement
- un titre
- une rubrique (Culture, Sport...)
- une ville, un quartier et éventuellement un lieu-dit (exemple: lieu unique)
- une adresse
- une date et une heure
- la possibilité de consulter le site web de l'événement
- la possibilité de consulter Google Map pour localiser l'événement
- Nous avons choisi d'utiliser un **code couleur** pour chaque événement:
  - Vert si l'événement a lieu normalement
  - Orange si l'événement est reporté
  - Rouge si l'événement est annulé

A noter qu'il arrive qu'un événement ait la mention reporté ou annulé dans son titre mais ne soit pas affiché en orange et rouge. Cela est dû au fait que nous nous sommes basé sur les données "reporte" ou "annule" du JSON fourni par l'API et non sur le titre de l'événement.

- Il est aussi possible d'effectuer une **recherche d'un événement** sur son nom via la barre de recherche en haut de la page Agenda des événements.
- L'application est **responsive**. Les "cards" d'événements s'adaptent de la manière suivante si l'on reprend la classe Bulma: `class="column is-half-tablet is-one-third-desktop is-one-quarter-widescreen"`

## Plugins Capacitors:

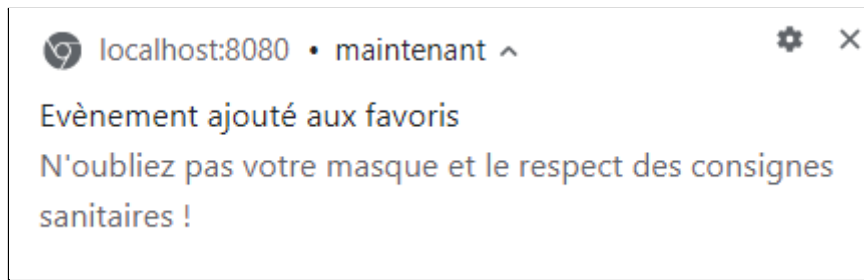
Nous avons utilisé les plugins Capacitor suivants dans notre application Agenda:

- Le plugin **Geolocation**, qui permet de récupérer la position de l'utilisateur. Nous avons repris l'exemple du cours pour pouvoir afficher, pour chaque événement, la distance en km entre le lieu de l'événement et le lieu où l'utilisateur se trouve. Nous avons adapté cet exemple pour qu'il corresponde au format des coordonnées géographiques renseignées dans l'API où l'on récupère les informations.

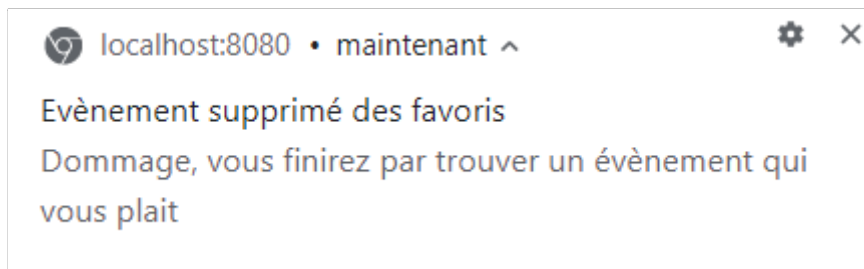


*Affichage de la proximité d'un événement avec le plugin Capacitor*

- Le plugin **LocalNotification** permettant d'afficher un message sous forme de notification à l'utilisateur. Ces notifications apparaissent quand un utilisateur ajoute ou supprime un favori sur l'application.



*Notification dans le cadre d'ajout de favoris avec le plugin LocalNotification*



*Notification dans le cadre de la suppression de favoris avec le plugin LocalNotification*

## **Conclusion :**

Ce projet nous a permis de prendre en main le développement d'une application avec VueJS. Cette première expérience a été bénéfique pour notre binôme puisque nous avons pu mettre en pratique ce qui a été vu en cours et de mieux appréhender l'ensemble des notions. Nous avons cependant rencontré quelques difficultés durant ce projet, notamment sur la page des favoris avec l'alimentation du tableau contenant les événements à afficher (Difficulté résolue avec l'aide de Mr. Morel). De plus, nous avons pris du temps à trouver la meilleure solution pour garder en mémoire la liste des id des événements qui seront à afficher dans la page favoris. La solution choisie a été celle détaillée plus haut dans ce rapport.

Pour conclure sur ce projet, nous avons réussi à implémenter l'ensemble de nos idées pour l'application et nous avons beaucoup appris sur le développement via VueJS qui était une première pour nous.